October 1981

8203/8206/2164A Memory Design

Brad May
Peripherals Applications

order number:210315.001

8203/8206/2164A Memory Design

Contents

ABSTRACT

This Application Note shows an error corrected dynamic RAM memory design using the 8203 64K Dynamic RAM Controller, 8206 Error Detection and Correction Unit and 150 ns 64K Dynamic RAMs with a minimum of additional logic.

The goals of this design are to:

1. Control 128K words×16 bits (256 KB) of 64K dynamic RAM.
2. Support 150 ns dynamic RAMs.
3. Write corrected data back into dynamic RAM when errors are detected during read operations.
4. To use a minimum of additional logic.

It is not the goal of this design to:

1. Provide the maximum possible performance.
2. Provide features like error logging, automatic error scrubbing and dynamic RAM initialization on power-up, or diagnostics, although these features can be added.

DESIGN

Figure 1 shows a memory design using the 8206 with Intel's 8203 64K Dynamic RAM Controller and 150 ns 64K Dynamic RAMs. As few as three additional ICs complete the memory control function (Figure 2).

For simplicity, all memory cycles are implemented as single-cycle read-modify-writes, shown in Figure 3. This cycle differs from a normal read or write primarily when the dynamic RAM write enable ($\overline{WE}$) is activated. In a normal write cycle, $\overline{WE}$ is activated early in the cycle; in a read cycle, $\overline{WE}$ is inactive. A read-modify-write cycle consists of two phases. In the first phase, $\overline{WE}$ is inactive, and data is read from the dynamic RAM; for the second phase, $\overline{WE}$ is activated and the (modified) data is written into the same word in the dynamic RAM. Dynamic RAMs have separate data input and output pins so that modified data may be written, even as the original data is being read. Therefore data may be read and written in only one memory cycle.

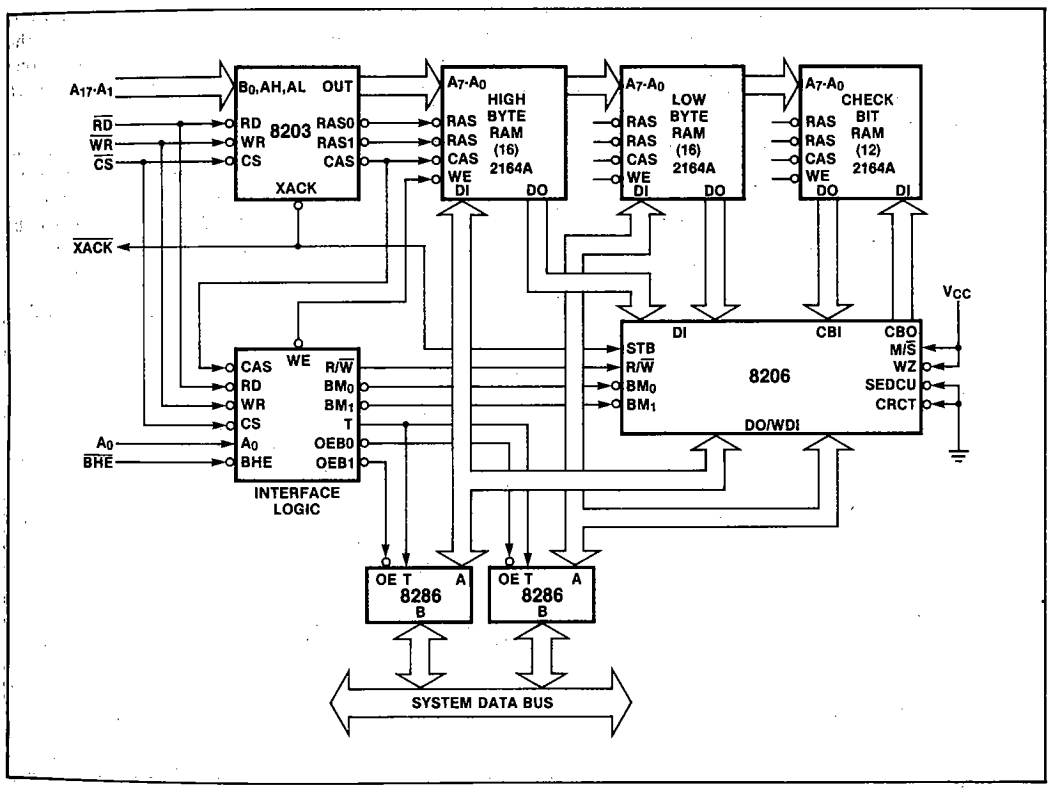

Figure 1. 8203/8206 Memory System

In order to do read-modify-writes in one cycle, the dynamic RAM's $\overline{CAS}$ strobe must be active long enough for the 8206 to access data from the dynamic RAM, correct it, and write the corrected data back into the dynamic RAM. $\overline{CAS}$ active time is an 8203 spec (t_{CAS}), and is dependent on the 8203's clock frequency. The clock frequency and dynamic RAM must be chosen to satisfy Equation 1.

(Eq. 1)

8203		Dynamic RAM		8206		8206		Dynamic RAM		Dynamic RAM
t_{CAS}min	≥	t_{CAC}	+	TDVQV	+	TQVQV	+	t_{DS}	+	t_{CWL}
5(54)−10	≥	85	+	67	+	59	+	0	+	40
260	≥	251		✓						

The 8203 itself performs normal reads and writes. In order to perform read-modify-writes, all that is needed is to change the timing of the $\overline{WE}$ signal. In this design, $\overline{WE}$ is generated by the interface logic in Figure 2—the 8203 $\overline{WE}$ output is not used. All other dynamic RAM control signals come from the 8203. A 20-ohm damping resistor is used to reduce ringing of the WE signal. These resistors are included on-chip for all 8203 outputs.

The interface logic generates the R/$\overline{W}$ input to the 8206. This signal is high for read cycles and low for write cycles. During a read-modify-write cycle, R/$\overline{W}$ is first high, then low. The falling edge of R/$\overline{W}$ tells the 8206 to latch its syndrome bits internally and generate corrected check bits to be written to dynamic RAM. Corrected data is already available from the DO pins. No control signals at all are required to generate corrected data.

R/$\overline{W}$ is generated by delaying $\overline{CAS}$ from the 8203 with a TTL-buffered delay line. This allows the 8206 sufficient time to generate the syndrome; this delay, $t_{DELAY\ 1}$, must satisfy Equation 2.

(Eq. 2)

		Dynamic RAM		8206
$t_{DELAY\ 1}$	≥	t_{CAC}	+	TDVRL
150	≥	85	+	34
150	≥	119		✓

The 8206 uses multiplexed pins to output first the syndrome word and then check bits. This same R/$\overline{W}$ signal may be used to latch the syndrome word externally for error logging. The 8206 also supplies two useful error signals. $\overline{ERROR}$ signals the presence of an error in the data or check bits. CE tells if the error is correctable (single bit in error) or uncorrectable (multiple bits in error).

In the event that an uncorrectable error is detected, the 8206 will force the Correctable Error (CE) flag low; this may be used as an interrupt to the CPU to halt execution and/or perform an error service routine. In this case the 8206 outputs data and check bits just as they were read, so that the data in the dynamic RAM is left unaltered, and may be inspected later.

After R/$\overline{W}$ goes low, sufficient time is allowed for the 8206 to generate corrected check bits, then the interface logic activates $\overline{WE}$ to write both corrected data and check bits into dynamic RAM. $\overline{WE}$ is generated by delaying CAS from the 8203 with the same delay line

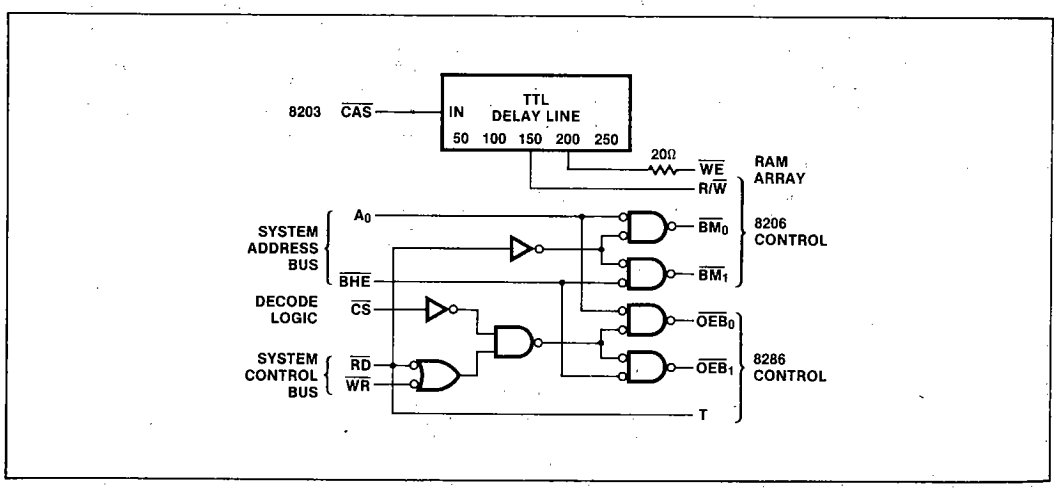

Figure 2. Interface Logic

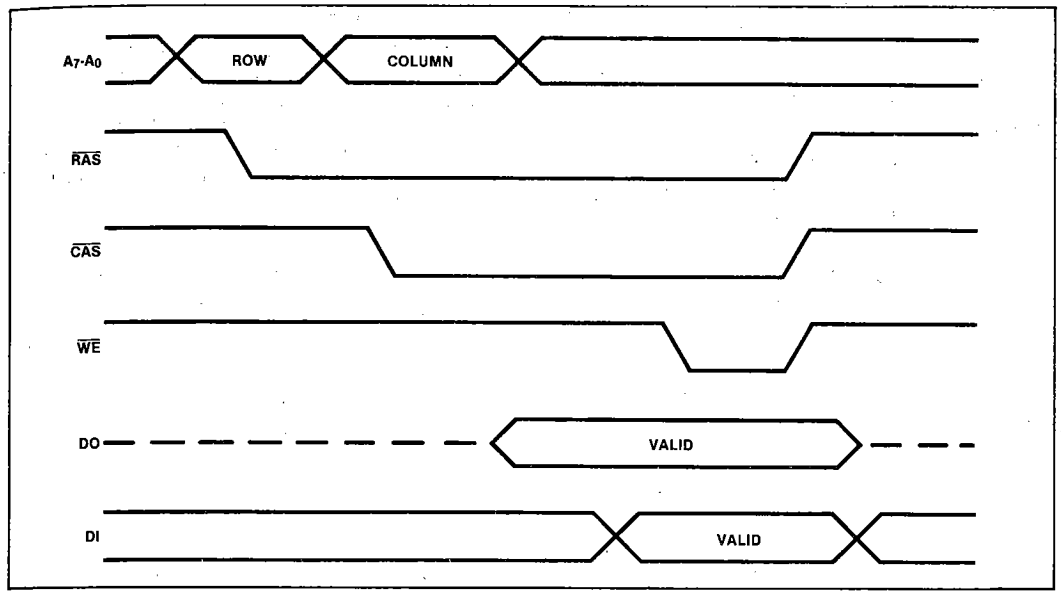

Figure 3. Single-Cycle Read-Modify-Write

used to generate R/$\overline{W}$. This delay, $t_{DELAY\ 2}$, must be long enough to allow the 8206 to generate valid check bits, but not so long that the t_{CWL} spec of the RAM is violated. This is expressed by Equation 3.

(Eq. 3)

		8206				8203		Dynamic RAM
$t_{DELAY\ 1}$	+	TRVSV	≤	$t_{DELAY\ 2}$	≤	t_{CAS}min	−	t_{CWL}
150	+	42	≤	200	≤	260	−	40
	192		≤	200	≤	220		✓

Unlike other EDC chips, errors in both data *and check bits* are automatically corrected, without programming the chip to a special mode.

Since the 8203 terminates $\overline{CAS}$ to the dynamic RAMs a fixed length of time after the start of a memory cycle, a latch is usually needed to maintain data on the bus until the 8086 completes the read cycle. This is conveniently done by connecting $\overline{XACK}$ from the 8203 to the STB input of the 8206. This latches the read data and check bits using the 8206's internal latches.

The 8086, like all 16-bit microprocessors, is capable of reading and writing single byte data to memory. Since the Hamming code works only on entire words, if you want to write one byte of the word, you have to read the entire word to be modified, do error correction on it, merge the new byte into the old word inside the 8206, generate check bits for the new word, and write the whole word plus check bits into dynamic RAM. A byte write is implemented as a Read-Modify-Write.

Why bother with error correction on the old word? Suppose a bit error had occurred in the half of the old word not to be changed. This old byte would be combined with the new byte, and new check bits would be generated for the whole word, *including the bit in error*. So the bit error now becomes "legitimate"; no error will be detected when this word is read, and the system will crash. You can see why it is important to eliminate this bit error before new check bits are generated. Byte writes are difficult with most EDC chips, but easy with the 8206.

Referring again to Figure 2, the 8206 byte mark inputs ($\overline{BM_0}$, $\overline{BM_1}$), are generated from A0 and $\overline{BHE}$, respectively, of the 8086's address bus, to tell the 8206 which byte is being written. The 8206 performs error correction on the entire word to be modified, but tri-states its DO/WDI pins for the byte to be written; this byte is provided from the data bus by enabling the corresponding 8286 transceiver. The 8206 then generates check bits for the new word.

During a read cycle, $\overline{BM_0}$ and $\overline{BM_1}$ are forced inactive, i.e., the 8206 outputs both bytes even if 8086 is only reading one. This is done since all cycles are implemented as read-modify-writes, so both bytes of data (plus check bits) must be present at the dynamic RAM data input pins to be rewritten during the second phase of the read-modify-write. Only those bytes actually be-

ing read by the 8086 are driven on the data bus by enabling the corresponding 8286 transceiver.

The output enables of the 8286 transceivers ($\overline{OEB0}$, $\overline{OEB1}$) are qualified by the 8086 $\overline{RD}$, $\overline{WR}$ commands and the 8203 $\overline{CS}$. This serves two purposes:

1. It prevents data bus contention during read cycles.
2. It prevents contention between the transceivers and the 8206 DO pins at the beginning of a write cycle.

CONCLUSION

Thanks to the use of a 68-pin package, the 8206 Error Detection and Correction Unit is able to implement an architecture with separate 16 pin input and output busses. The resulting simplification of control requirements allows error correction to be easily added to an 8203 memory subsystem with a minimal amount of interface logic.